水下鱼类品种识别模型与实时识别系统

李少波[1,2,3]，杨　玲[1,2,3]，于辉辉[4]，陈英义[1,2,3*]

（1. 中国农业大学 信息与电气工程学院，北京 100083；2. 中国农业大学国家数字渔业创新中心，北京 100083；3. 北京市农业物联网工程技术研究中心，北京 100083；4. 北京林业大学 信息学院，北京 100083）

摘　要：快速准确的鱼类识别系统需要良好的识别模型和部署系统作为支撑。近年来，卷积神经网络在图像识别领域取得了巨大成功，不同的卷积网络模型都有不同的优点和缺点，面对众多可供选择的模型结构，如何选择和评价卷积神经网络模型成为了必须考虑的问题。此外，在模型应用方面，移动终端直接部署深度学习模型需要对模型进行裁剪、压缩处理，影响精度的同时还会导致安装包体积增大，不利于模型升级维护。针对上述问题，本研究根据水下鱼类实时识别任务特点，选取了AlexNet、GoogLeNet、ResNet和DenseNet预训练模型进行对比试验研究，通过在Ground-Truth鱼类公开数据集基础上对图像进行随机翻转、旋转、颜色抖动来增强数据，使用Label smoothing作为损失函数缓解模型过拟合问题，通过研究Ranger优化器和Cosine学习率衰减策略进一步提高模型训练效果。统计各个识别模型在训练集和验证集上的精确度和召回率，最后综合精确度和召回率量化模型识别效果。试验结果表明，基于DenseNet训练的鱼类识别模型综合评分最高，在验证集的精确度和召回率分别达到了99.21%和96.77%，整体F_1值达到了0.9742，模型理论识别精度达到预期。基于Python开发并部署了一套远程水下鱼类实时识别系统，将模型部署到远程服务器，移动终端通过网络请求进行鱼类识别模型调用，验证集图像实际测试表明，在网络良好条件下，移动终端可以在1 s内准确识别并显示鱼类信息。

关键词：鱼类识别模型；卷积神经网络；模型评价；安卓；Ground-Truth；实时识别系统

中图分类号：S964;TP183　　　　**文献标志码：**A　　　　**文章编号：**SA202202006

引用格式：李少波，杨玲，于辉辉，陈英义．水下鱼类品种识别模型与实时识别系统[J]. 智慧农业(中英文), 2022, 4(1): 130-139.

LI Shaobo, YANG Ling, YU Huihui, CHEN Yingyi. Underwater fish species identification model and real-time identification system[J]. Smart Agriculture, 2022, 4(1): 130-139. (in Chinese with English abstract)

1　引　言

鱼类是最重要的水产品之一，世界上已知的鱼类有28,000多种[1]，研发快速、准确的自动化鱼类识别系统在鱼类知识科普、混合养殖、海洋监管等方面具有重要意义。一个完整的自动化鱼类识别系统至少应该包含识别方法的构建和模型部署两个方面。

在鱼类识别方法方面，传统的基于手工特征提取的鱼类识别方法，主要依靠鱼的纹理、颜色、形状等特征进行区分[2]，虽然通过应用尺度不变特征变换（Scale-Invariant Feature Trans-

收稿日期：2021-09-07

基金项目：广东省科技计划项目（2017B010126001）；国家自然科学基金项目（62076244）；国家重点研发计划（2017YFE0122100）

作者简介：李少波（1993—），男，硕士研究生，研究方向为计算机应用技术。Email：279173135@qq.com。

*通信作者：陈英义（1980—），男，博士，教授。研究方向为农业信息化技术、人工智能技术与空间信息技术。E-mail：chenyingyi@cau.edu.cn。

form，SIFT）[3]、方向梯度直方图（Histogram of Oriented Gradient，HOG）[4] 等方法取得了卓有成效的改进，但鱼类特征需要依赖专家手动设计，当识别另一种鱼类时需要重新设计这些特征，方法不具有通用性且识别精度不高 [5]。近年来，卷积神经网络（Convolution Neural Network，CNN）在图像识别领域取得了巨大进展 [6]，Krizhevsky 等 [7] 提出的 AlexNet 卷积神经网络在 ImageNet Large Scale Visual Recognition Challenge（ILSVRC）比赛中大放异彩，一举斩获了 ILSVRC2012 比赛冠军。在 AlexNet 之后，研究者们陆续提出了许多优秀的改进的 CNN 模型，如 VGGNet [8]、GoogLeNet [9]、ResNet [10]、DenseNet [11] 等，这些模型在解决图像分类问题上都取得了优异的成绩。CNN 系列算法已经成为了完成图像识别任务的最佳算法之一 [12]。不同的 CNN 模型有不同的特点，没有一种模型可以在所有场景下都优于其它模型 [13]，例如 VGGNet 相比 AlexNet 可以学习更多特征，但也大幅增加了模型大小 [14]；GoogLeNet 的 Inception 模块大幅减少了模型参数，但并没有解决随着深度的增加可能会出现的模型退化问题 [15]；ResNet 缓解了随着深度增加导致的梯度消失问题，但增加了实现难度并且引入了许多需要手工设置的权重 [16]。对于具体的水下鱼类识别任务，CNN 结构如何选择和评价模型成为必须考虑的问题。

在模型部署方面，将模型直接部署到安卓移动终端是一种可行的方案。随着移动互联网时代发展和智能终端的普及，越来越多的人习惯用手机上网。《中国互联网络发展状况统计报告》显示，截止到 2020 年 12 月，中国手机网民规模达到了 9.86 亿，网民中使用手机上网的比例达到了 99.7% [17]。2021 年安卓系统占据了将近 90% 手机市场份额 [18]，在安卓终端上实现鱼类识别可方便用户使用，大幅降低终端用户学习成本。但安卓终端相比于服务器计算能力有限，一般的深度模型需要针对移动设备进行裁剪、压缩处理后才能部署到移动终端，对模型进行裁剪和压缩会影响模型精度，而且将模型直接打包到安装包会显著影响安装包大小，不利于功能扩展和维护升级。

针对上述问题，本研究以实现高精度鱼类识别系统为出发点，尝试通过研究多个主流 CNN 结构，结合数据集特点和任务要求初步选择模型，通过设计模型对比试验进一步量化模型优劣，并设计一套远程鱼类识别系统完成模型部署，以实现一套完整的高精度鱼类在线识别系统。

2 研究数据与方法

2.1 系统设计

鱼类识别系统的整体设计如图 1 所示，用户可以通过调用安卓手机摄像头采集鱼类图像信息，通过网络上传鱼类图像至服务器进行识别，服务器在成功识别鱼类品种后查询鱼类信息数据库获取鱼的简介、分布、习性、生长周期等鱼类知识库信息返回给安卓客户端显示。

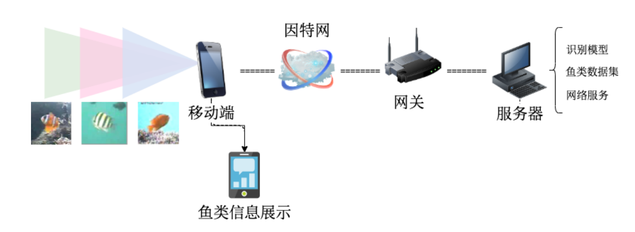


图 1　鱼类识别系统架构示意图

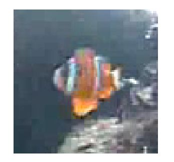
Fig. 1 Architecture of fish identification system

2.2 数据处理

采用 Ground-Truth 公开数据集 [19] 中图像训练模型。数据集包含由海洋生物学家手动标记的 23 种鱼类共计 27,370 张水下图像，图 2 展示了数据集的每种鱼类的图像和数量。将图像分成 80% 的训练数据集（21,819 张图像）和 20% 的验证数据集（5474 张图像），在训练过程中使用随机水平翻转、随机旋转、颜色抖动等数据增强技术扩大样本大小。图像大小缩放到 260×260 像素作为模型输入。


五带豆娘鱼98张


双斑刺尾鱼218张


双带小丑鱼4049张

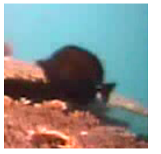

长棘光鳃鱼3593张

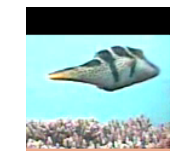

黄足笛鲷206张

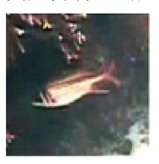

褐蓝子鱼25张

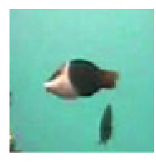

弓月蝴蝶鱼2534张

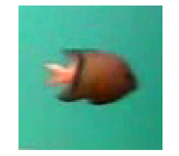

双线眶棘鲈49张

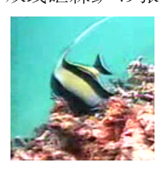

黑三角倒吊鱼90张


黑马鞍鲀鱼147张


莎姆金鳞鱼299张

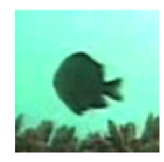

黑鳍粗唇鱼42张


迪克氏固曲齿鲷2683张

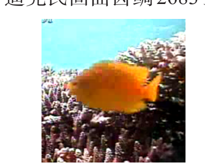

镰鱼21张


黑嘴雀鱼16张

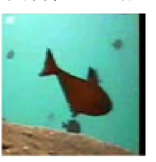

鹦嘴鱼56张


网纹宅泥鱼12,112张


康德锯鳞鱼450张


摩鹿加雀鲷181张


黄纹炮弹鱼41张


黑缘单鳍鱼29张


横带粗唇鱼241张

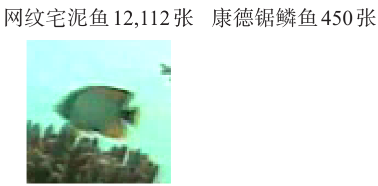

川纹蝴蝶鱼190张

图2 Ground-Truth数据集的图像及样本量

Fig. 2 The images and quantity of Ground-Truth dataset

2.3 基本原理

2.3.1 卷积神经网络模型

卷积神经网络是图像处理领域广泛应用的深度学习网络，本质上是一种层次网络[20]，其主要的特点是包含卷积层和池化层，原始输入图像的具象信息通过层层卷积和池化操作被提取成高层次抽象信息[21]，到最后一层后根据损失函数计算预测值和真实结果的误差再逐层向前传播矫正模型参数，直到模型收敛[22]。卷积和最大池化操作如图3所示。

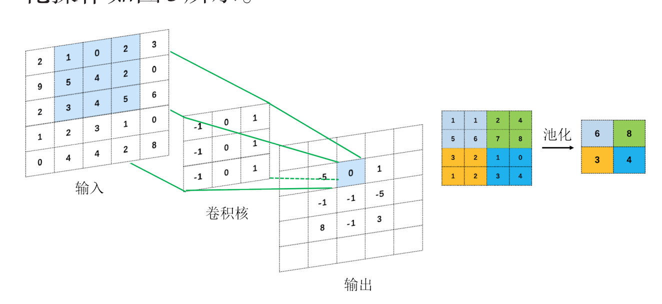


图3 卷积和最大池化

Fig.3 Convolution and max pooling

卷积操作的计算公式为[23]：

$$X_j^L = f\left(\sum_{i \in M_j} X_i^{L-1} \times K_{i,j}^L + b_j^L\right) \quad (1)$$

其中，L表示神经网络层数；M_j表示输入特征图；$K_{i,j}^L$表示卷积核；b_j^L表示偏置；$f(\cdot)$表示激活函数。

AlexNet、VGGNet、GoogLeNet、ResNet、DenseNet是CNN模型的典型代表，统计显示[24]，各个CNN模型大小如表1所示。

表1 5种CNN模型大小

Table 1 The sizes of the five kinds of CNNs

CNN名称	模型大小/Mb
AlexNet	60
VGGNet	138
GoogLeNet	4
ResNet	1.7～25.6
DenseNet	15.3～25.6

2.3.2 混淆矩阵

混淆矩阵是一种比较预测类别和真实类别差异的可视化矩阵[25]。每一列的总数表示预测为该类别的数据的数目，每一行的数据总数表示该

类别的数据实际的数目，二分类任务的混淆矩阵如图4所示。

混淆矩阵		真实值	
		Negative	Positive
预测值	Negative	TN	FN
	Positive	FP	TP

图4　二分类混淆矩阵

Fig. 4 Confusion matrix of binary classification

其中，TN（True Negative）表示预测为Negative类正确的数量，FN（False Negative）表示预测为Negative类错误的数量，FP（False Positive）表示预测为Positive类错误的数量，TP（True Positive）表示预测为Positive类正确的数量。

在混淆矩阵的基础上延伸出了精确率（Precision）、召回率（Recall）、准确率（Accuracy）等指标，精确率的计算公式为：

$$Precision = \frac{TP}{TP + FP} \quad (2)$$

召回率的计算公式为：

$$Recall = \frac{TP}{TP + FN} \quad (3)$$

准确率的计算公式为：

$$Accurary = \frac{TP + TN}{TP + TN + FP + FN} \quad (4)$$

平衡分数F_1值是综合考虑了精确度、召回率的调和值，一般F_1值越大说明模型效果越好，其计算公式为：

$$F_1 = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (5)$$

2.4 模型选择与构建

CNN模型要想获得较理想的识别模型，需要基于大量数据集训练[26]，而且识别任务越复杂需要的数据集越大[27]。对于水下鱼类实时识别任务，由于光线的变换、水的折射等复杂水下环境，水下鱼类图像往往存在分辨率不足、噪点多等特点[2]，而且不同种类的鱼的生存水域和生活习性存在巨大差异，一些鱼类只能生活在深海之中，获取大规模的水下鱼类图像并不容易。由于可获取的水下鱼类数据集比较有限，而AlexNet、ResNet等著名CNN模型已经在图像分类问题上经过了长时间、大规模的训练，以这些预训练模型为起点迁移到新的训练任务的迁移学习方法可以有效降低模型对于训练集大小的依赖，适合计算资源紧张、数据集不多的情形[28]，本试验采用基于预训练模型迁移学习的方法生成模型。从表2可以看出，VGGNet生成的模型最大，达到了138 Mb，模型越大意味着模型计算量越大，从而识别速度越慢。由于本次鱼类实时识别任务对模型识别速度要求较高，本试验选择了AlexNet、GoogLeNet、ResNet和DenseNet预训练网络来训练鱼类识别模型，通过对各个模型最后的全连接层进行微调适配水下鱼类识别任务。

为缓解模型过度拟合，选择Label smoothing[29]作为损失函数。Ranger[30]优化器结合了Radam[31]和Lookahead[32]到单一的优化器中，从而可以获取更高的精度，本文选择了Ranger作为训练优化器。将学习率统一设置成0.0001，并使用Cosine策略衰减学习率，批处理大小为64，迭代30轮，完整的超参数的配置如表2所示。

表2　鱼类识别试验超参数配置

Table 2 Super parameter configuration of fish identification

超参数	值
训练轮数/轮	30
批处理大小	64
学习率	0.0001
优化器	Ranger
学习率衰减	Cosine
损失	Label Smoothing

3　结果分析

根据表2参数设置，分别在AlexNet、GoogLeNet、ResNet50和DenseNet169上进行模型训练，并在验证集上测试了模型准确率、精确率、召回率等，试验结果如表3所示。

表3 鱼类识别试验不同卷积神经网络模型的对比

Table 3 Comparison of different convolutional neural network models for fish recognition experiment

模型	训练集精度	验证集表现			模型参数量/百万	浮点计算次数/G
		准确度/%	精确度/%	召回率/%		
AlexNet	99.42	99.08	95.59	93.76	61.10	0.93
GoogLeNet	99.81	99.51	98.37	94.75	5.62	1.98
ResNet50	99.83	99.58	99.26	96.64	23.56	5.98
DenseNet169	99.90	99.64	99.21	96.77	14.15	4.49

准确率可以很大程度上说明模型的优劣，对于一般的分类问题，我们希望模型越准确越好。由图1可以看出，本试验所采用的Ground-Truth数据集存在严重的数据不平衡问题，最少的黑嘴雀鱼只有16张图片，而最多的网纹宅泥鱼则多达12,112张图片，相差750多倍。此种情况下，模型只需要将分类结果分成网纹宅泥鱼就会得到很高的准确率，仅依靠准确率指标已经无法准确评估模型效果。由表3可以看出基于GoogLeNet训练的模型参数量最小，AlexNet模型的浮点计算量最小但同时参数量最大， DenseNet169的模型参数量和浮点计算次数比较均衡，基于ResNet50训练的模型验证集精确度最高达到了99.26%，但同时召回率低于DenseNet169。通常精确率越高，召回率就会越低。由公式（5）可以看到F_1值能够更全面地衡量模型优劣，更适合用于评估本试验模型。根据表3的结果，进一步计算各个模型整体的F_1值，结果如表4所示。

表4 鱼类识别试验不同卷积神经网络模型的F_1值

Table 4 F_1 values of different convolutional neural network models for fish recognition tests

模型	整体F_1值
AlexNet	0.9449
GoogLeNet	0.9537
ResNet50	0.9739
DenseNet169	0.9742

由表4可见， DenseNet169模型整体F_1值达到了0.9742，是本次试验获得的F_1值最高的模型。虽然基于GoogLeNet训练的模型参数量和浮点计算次数都少于DenseNet169模型，但考虑模型是部署到有较强计算能力的服务器端，并且优先保障鱼类识别应用的精准度，本研究选择了识别精度更高的DenseNet169模型进行部署。

4 水下鱼识别模型部署

4.1 服务器实现

（1） 开发平台

操作系统：Ubuntu 20.04。

平台环境：Python 3.8，CUDA 11.2，PyTorch 1.7.0，Anaconda 2.0.3，Django 3.2.4 [33]，Gunicorn 20.0.4 [34] 等。

开发工具：PyCharm Professional 2019.2。

数据库：MySQL。

为方便模型调用，本研究选择了Python语言进行服务器开发，在配备GT1650显卡的Ubuntu系统上安装了英伟达显卡驱动，在Anaconda创建虚拟环境并在虚拟环境中安装Django、PyTorch、numpy、pandas等必要工具包完成环境配置。

（2） 项目搭建

在PyCharm Professional中新建Django项目，并将Python解释器配置成上述虚拟环境，按照Django框架规范在配置文件里配置好数据库和安全策略等各类中间件，创建FishInfo类映射到数据库并纳入Django Admin管理，在urls.py文件中完成URL映射，服务器项目结构如图5所示。

（3） 模型集成

参考Pytorch官方文档 [35] 将上述权重文件打包成*.tar格式，创建classifier类，在该类的__init__方法中使用Pytorch提供的torch系列API完成模型初始化，并新建perdict函数实现模型预测逻辑，最终通过在Django的views.py中引入classifier类，传递图像数据给classifier对象完成

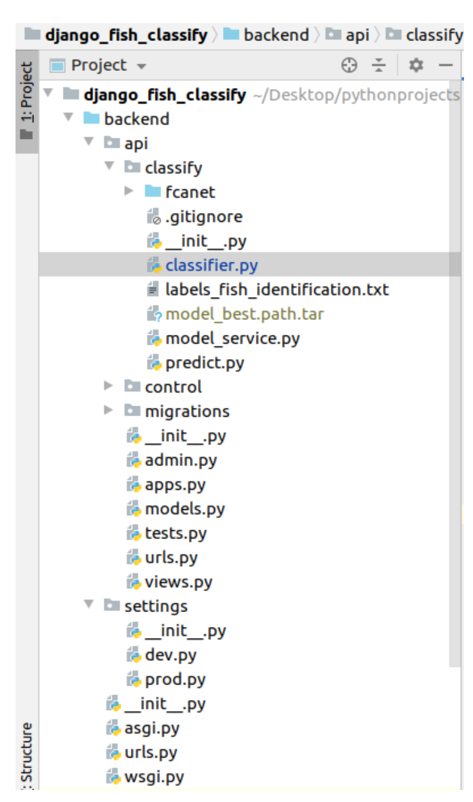


图5 鱼类识别系统服务器项目结构

Fig. 5 Server project structure of the fish identification system

模型调用，模型集成和初始化代码如图6所示。

```
class classifier(object):
    # 定义构造方法模型初始化
    def __init__(self):
        resume = os.path.dirname(__file__) + '/model_best.path.tar'
        torch.cuda.set_device(0)
        checkpoint = torch.load(resume)
        from collections import OrderedDict
        new_state_dict = OrderedDict()
        state_dict = {k: v for k, v in checkpoint.items()}
        for k, v in state_dict.items():
            print(k)
            if(k=="state_dict"):
                for a, b in v.items():
                    name = a[7:]
                    new_state_dict[name] = b
        self.model = models.densenet169(pretrained=True)
        self.model.fc = torch.nn.Linear(2048, 23)
        for parma in self.model.parameters():
            parma.requires_grad = False
        self.model.load_state_dict(new_state_dict, strict=False)
        self.model = self.model.cuda()
        self.model.eval()
        torch.no_grad()
```

图6 鱼类识别系统初始化识别模型

Fig. 6 Initialize the identification model of the fish identification system

（4） 鱼类信息存储

本系统采用MySQL来存储结构化的鱼类信息，系统所涉及的核心表为FishInfo鱼类信息表，包含了Ground-Truth数据集中鱼的介绍性信息，这些鱼类信息采集自乐潜无线海洋鱼类资料库[36]和台湾鱼类资料库[37]，FishInfo表的设计如表5所示。

表5 鱼类识别系统FishInfo表设计

Table 5 Design of FishInfo table of the fish identification system

字段名	字段类型	完整性约束	是否为主键	字段含义
fish_id	int	非空	是	鱼类编号
type	varchar	非空	否	鱼的种类
profile	varchar	非空	否	简介信息
outline	varchar	非空	否	外形介绍
distribution	varchar	非空	否	分布情况
life_cycle	varchar	非空	否	生长周期
disease	varchar	非空	否	常见病害

（5） 服务发布

由于Django本身带的调试服务器性能较弱，为提高服务器并发效率，本研究使用Gunicorn[34]多线程形式来发布服务端程序，最终服务器的相关环境和提供的接口服务如图7所示。

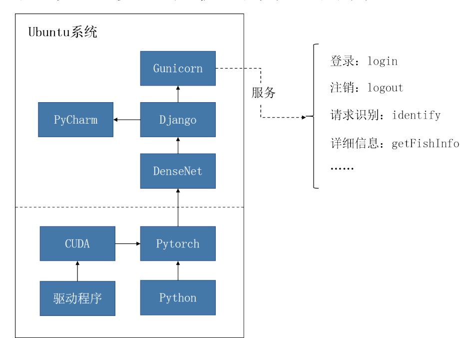


图7 鱼类识别系统服务器环境配置图

Fig. 7 Server environment configuration diagram of the fish identification system

4.2 安卓APP实现

（1） 开发平台

操作系统：macOS 11.2。

平台环境：JDK 1.8.0_211，Android SDK Platform 30。

开发工具：AndroidStudio 4.1.2[38]。

在Mac电脑上安装JDK、Android SDK并配置好环境变量，在AndroidStudio IDE中完成具体的APP程序编码工作。

（2） 项目搭建

为提高编码效率，APP引入了OKHttp[39]轻量级网络框架，在子线程中收发识别请求，由于谷歌规定安卓代码在子线程中不能更新界面，为方便界面交互引入了EventBus[40]来完成跨线程事件传递。以上两个依赖均为开源项目依赖，在AndroidStudio的build.gradle配置文件中添加相应的仓库地址即可完成依赖项导入。APP项目结构如图8所示。

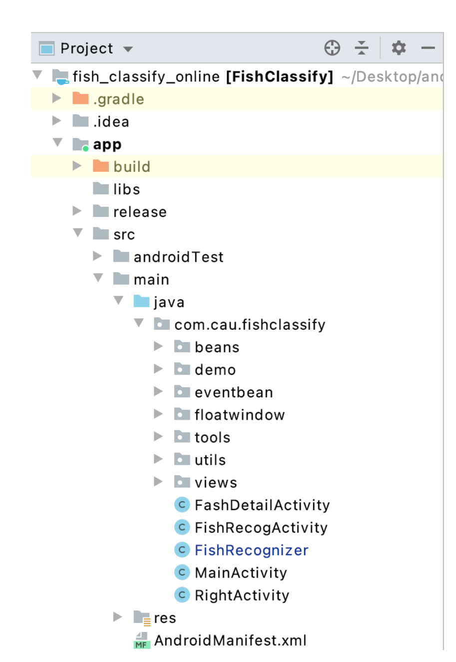


图8　鱼类识别系统安卓端APP项目结构

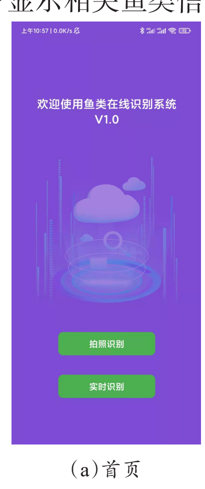
Fig. 8　Android APP project structure of the fish recognition system

（3） 图像采集和上传

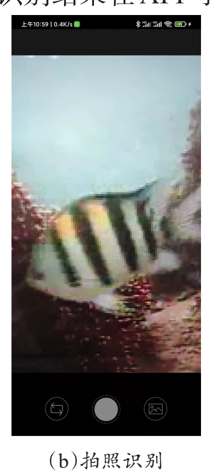
在图像采集方面，为尽可能提高软件兼容性，APP锁定屏幕为竖屏，采用Android SDK提供的Camera类完成相机的调用和图像帧获取，将原NV21格式图像帧包装成Bitmap并压缩为固定的640×480像素大小，最终编码成Base64格式字符串，通过OKHttp在子线程中将Base64编码的图片上传至服务器请求识别。

（4） 界面实现

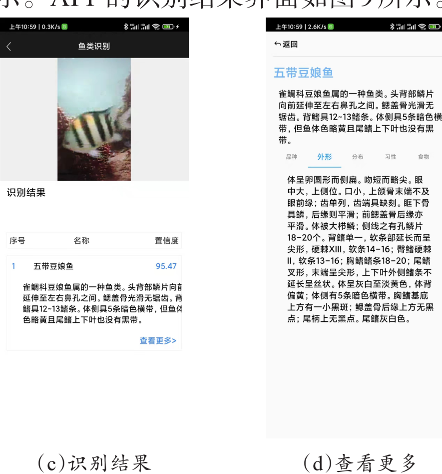
当服务器放回的识别可信度达到一定阈值后，APP显示相关鱼类信息，识别结果在APP子线程中接收，通过EventBus完成识别结果的事件分发，最终在主线程的界面完成识别结果显示。APP的识别结果界面如图9所示。



（a）首页



（b）拍照识别



（c）识别结果　（d）查看更多

图9　安卓版鱼类在线识别移动APP界面

Fig.9　Mobile APP interfaces of fish identification

4.3 系统测试

（1） 测试环境

软件环境：鱼类在线识别系统V1.0。

移动终端：红米K40手机。

服务器环境：配备 Intel i7-8700 处理器和英伟达GTX1650显卡的联想台式机。

网络环境：校园无线局域网。

场地环境：实验室环境40 W荧光灯下。

（2） 测试方法

通过在程序中加入耗时计算代码测试识别过

程耗时，计量模型识别耗时以及从手机终端发出网络请求到收到识别信息的总时延，具体的测试操作如图10所示，在红米K40手机上安装软件后，在台式电脑显示器上打开验证集中的鱼类图片，利用手机后置摄像头拍照进行鱼类识别，手机软件上实时显示识别结果和识别耗时。



图10　鱼类识别系统测试示意图

Fig.10 Schematic diagram of the fish identification system test

（3）　测试结果

使用上述环境和方法对鱼识别系统应用的实际测试，测试结果如表6所示。由表6可见，在上述测试环境下，APP可以在百毫秒级准确识别鱼类图片，并且延迟的主要原因是网络传输耗时，基本可以满足识别应用需要。

5　结论与展望

本研究以水下鱼类实时识别任务为切入点，提供了一个完整的鱼类识别任务模型选择、训练的评价的思路，并提出了一种基于安卓移动终端的远程鱼类识别解决方案，该方案具有良好的扩展性，可以为其它图像识别任务提供重要参考，所实现系统可以应用于鱼类知识科普、鱼类养殖等各类需要识别鱼类的场景。本次试验结果表明：（1）基于DenseNet169训练的模型整体 F_1 值达到了0.9742，是本试验获得的最优模型。（2）本试验提出的基于安卓移动终端的远程鱼类识别解决方案取得了良好的识别效果，具有较好的识别实时性。

后续研究还可以在模型压缩裁剪领域展开工作，对所部署模型进行裁剪压缩，进一步提高模型识别效果。

表6　鱼类在线识别系统测试结果

Table 6 Test results of the fish online recognition system

鱼种类	可信度/%	识别耗时/s	网络耗时/s	总耗时/s
五带豆娘鱼	98.47	0.074	0.135	0.209
双斑刺尾鱼	97.22	0.060	0.108	0.168
双带小丑鱼	99.62	0.058	0.128	0.186
长棘光鳃鱼	97.01	0.059	0.137	0.196
黄足笛鲷	97.20	0.058	0.077	0.135
褐蓝子鱼	78.66	0.056	0.118	0.174
弓月蝴蝶鱼	95.97	0.059	0.105	0.164
双线眶棘鲈	85.08	0.071	0.115	0.186
黑三角倒吊鱼	99.39	0.061	0.107	0.168
黑马鞍鲀鱼	99.30	0.058	0.138	0.196
莎姆金鳞鱼	85.23	0.072	0.191	0.263
黑鳍粗唇鱼	95.52	0.062	0.113	0.175
迪克氏固曲齿鲷	91.77	0.061	0.111	0.172
镰鱼	88.06	0.060	0.099	0.159
黑嘴雀鱼	87.49	0.058	0.131	0.189
鹦嘴鱼	97.98	0.059	0.112	0.171
网纹宅泥鱼	96.69	0.064	0.079	0.143
康德锯鳞鱼	87.64	0.057	0.100	0.157
摩鹿加雀鲷	92.97	0.065	0.114	0.179
黄纹炮弹鱼	87.25	0.059	0.159	0.218
黑缘单鳍鱼	94.46	0.061	0.117	0.178
横带粗唇鱼	99.46	0.064	0.128	0.192
川纹蝴蝶鱼	91.46	0.057	0.105	0.162
平均值	**93.21**	**0.061**	**0.119**	**0.180**

参考文献：

[1] NELSON J S, GRANDE T C, WILSON M V. Fishes of the world[M]. New Jersey: John Wiley & Sons, 2016.

[2] YANG L, LIU Y, YU H, et al. Computer vision models in intelligent aquaculture with emphasis on fish detection and behavior analysis: A review[J]. Archives of Computational Methods in Engineering, 2021, 28(4): 2785-2816.

[3] LOWE D G. Distinctive image features from scale-invariant keypoints[J]. International journal of computer vision, 2004, 60(2): 91-110.

[4] DALAL N, TRIGGS B. Histograms of oriented gradients for human detection[C]// 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05). Piscataway, New York, USA: IEEE, 2005: 886-893.

[5] 卢宏涛, 张秦川. 深度卷积神经网络在计算机视觉中的应用研究综述[J]. 数据采集与处理, 2016, 31(1): 1-17.
LU H, ZHANG Q. Applications of deep convolutional neural network in computer vision[J]. Data Acquisition and Processing, 2016, 31(1): 1-17.

[6] AL-SAFFAR A A M, TAO H, TALAB M A. Review of deep convolution neural network in image classification[C]// 2017 International Conference on Radar, Antenna, Microwave, Electronics, and Telecommunications (ICRAMET). Piscataway, New York, USA: IEEE, 2017: 26-31.

[7] KRIZHEVSKY A, SUTSKEVER I, HINTON G E. Imagenet classification with deep convolutional neural networks[J]. Advances in Neural Information Processing Systems, 2012, 25: 1097-1105.

[8] SIMONYAN K, ZISSERMAN A. Very deep convolutional networks for large-scale image recognition[J/OL]. arXiv: 1409.1556, 2014.

[9] SZEGEDY C, LIU W, JIA Y, et al. Going deeper with convolutions[C]// The IEEE Conference on Computer vision and Pattern Recognition. Piscataway, New York, USA: IEEE, 2015: 1-9.

[10] HE K, ZHANG X, REN S, et al. Deep residual learning for image recognition[C]// The IEEE Conference on Computer Vision and Pattern Recognition. Piscataway, New York, USA: IEEE, 2016: 770-778.

[11] HUANG G, LIU Z, VAN DER MAATEN L, et al. Densely connected convolutional networks[C]// The IEEE Conference on Computer Vision and Pattern Recognition. Piscataway, New York, USA: IEEE, 2017: 4700-4708.

[12] HU H, YANG Y. A Combined GLQP and DBN-DRF for face recognition in unconstrained environments[C]// 2017 2nd International Conference on Control, Automation and Artificial Intelligence (CAAI 2017). Paris: Atlantis Press, 2017: 553-557.

[13] NOVAKOVIĆ J D, VELJOVIĆ A, ILIĆ S S, et al. Evaluation of classification models in machine learning[J]. Theory and Applications of Mathematics & Computer Science, 2017, 7(1): 39-46.

[14] ERTOSUN M G, RUBIN D L. Probabilistic visual search for masses within mammography images using deep learning[C]// 2015 IEEE International Conference on Bioinformatics and Biomedicine (BIBM). Piscataway, New York, USA: IEEE, 2015: 1310-1315.

[15] LIN C, LIN C, WANG S, et al. Multiple convolutional neural networks fusion using improved fuzzy integral for facial emotion recognition[J]. Applied Sciences, 2019, 9(13): ID 2593.

[16] BHATIA G S, AHUJA P, CHAUDHARI D, et al. Farmguide-one-stop solution to farmers[J/OL]. Asian Journal For Convergence In Technology, 2019, 4(1). [2021-09-06]. https://asianssr.org/index.php/ajct/article/view/789.

[17] 中国互联网络信息中心. 中国互联网络发展状况统计报告[EB/OL]. [2021-09-07]. http://www.cac.gov.cn/2021-02/03/c_1613923 423079314.htm.

[18] 孙彦博. 2021年中国手机操作系统行业研究报告[R]. 南京: 头豹研究院, 2021.

[19] BOOM B J, HUANG P X, HE J, et al. Supporting ground-truth annotation of image datasets using clustering[C]// The 21st International Conference on Pattern Recognition (ICPR2012). Piscataway, New York, USA: IEEE, 2012: 1542-1545.

[20] MAKANTASIS K, KARANTZALOS K, DOULAMIS A, et al. Deep supervised learning for hyperspectral data classification through convolutional neural networks[C]// 2015 IEEE International Geoscience and Remote Sensing Symposium (IGARSS). Piscataway, New York, USA: IEEE, 2015: 4959-4962.

[21] 魏秀参. 解析深度学习: 卷积神经网络原理与视觉实践[M]. 北京: 电子工业出版社, 2018: 13-14.

[22] RUMELHART D E, HINTON G E, WILLIAMS R J. Learning representations by back-propagating errors[J]. nature, 1986, 323(6088): 533-536.

[23] BOUVRIE J. Notes on convolutional neural networks[J/OL]. (2006-11-22) [2021-09-06]. https://www.researchgate.net/publication/28765140_Notes_on_Convolutional_Neural_Networks.

[24] KHAN A, SOHAIL A, ZAHOORA U, et al. A survey of the recent architectures of deep convolutional neural networks[J]. Artificial Intelligence Review, 2020, 53 (8): 5455-5516.

[25] RAHMAD F, SURYANTO Y, RAMLI K. Performance comparison of anti-spam technology using confusion matrix classification[C]// IOP Conference Series: Materials Science and Engineering. Bandung, Indonesia: IOP Publishing, 2020: 12076.

[26] WANG H, ZHAO T, LI L C, et al. A hybrid CNN feature model for pulmonary nodule malignancy risk differentiation[J]. Journal of X-ray Science and Technology, 2018, 26(2): 171-187.

[27] KAMILARIS A, PRENAFETA-BOLDÚ F X. A review of the use of convolutional neural networks in agriculture[J]. The Journal of Agricultural Science, 2018, 156(3): 312-322.

[28] 梁红, 金磊磊, 杨长生. 小样本情况基于深度学习的水下目标识别研究[J]. 武汉理工大学学报(交通科学与工程版), 2019, 43(1): 6-10.
LIANG H, JIN L, YANG C. Research on underwater target recognition based on depth learning with small sample[J]. Journal of Wuhan University of Technology (Transportation Science & Engineering), 2019, 43(1): 6-10.

[29] ZHANG C, JIANG P, HOU Q, et al. Delving deep into label smoothing[J]. IEEE Transactions on Image Processing, 2021, 30: 5984-5996.

[30] GitHub-lessw2020/Ranger-Deep-Learning-Optimizer: Ranger — A synergistic optimizer using RAdam (Rectified Adam), Gradient Centralization and LookAhead in one codebase[EB/OL]. [2021/8/30]. https://github.com/lessw2020/Ranger-Deep-Learning-Optimizer.
[31] LIU L, JIANG H, HE P, et al. On the variance of the adaptive learning rate and beyond[J/OL]. arXiv: 1908.03265, 2019.
[32] ZHANG M R, LUCAS J, HINTON G, et al. Lookahead optimizer: K steps forward, 1 step back[J/OL]. arXiv:1907.08610, 2019.
[33] Django overview[EB/OL]. [2021-07-28]. https://www.djangoproject.com/start/overview/.
[34] Gunicorn-Python WSGI HTTP Server for UNIX[EB/OL]. [2021-07-28]. https://gunicorn.org/.
[35] PyTorch documentation[EB/OL]. [2021-07-28]. https://pytorch.org/docs/stable/index.html.
[36] 中文海洋鱼类资料库[EB/OL]. [2021-08-30]. http://sea.fundiving.cn.
[37] 台湾鱼类资料库[EB/OL]. [2021-08-30]. https://fishdb.sinica.edu.tw/chi/species.php.
[38] Android developers[EB/OL]. [2021-07-28]. https://developer.android.google.cn/studio.
[39] OkHttp[EB/OL]. [2021-08-30]. https://square.github.io/okhttp/.
[40] GitHub-asaskevich/EventBus: [Go] Lightweight eventbus with async compatibility for Go[EB/OL]. [2021-08-30]. https://github.com/asaskevich/EventBus.

Underwater Fish Species Identification Model and Real-Time Identification System

LI Shaobo[1,2,3], YANG Ling[1,2,3], YU Huihui[4], CHEN Yingyi[1,2,3*]

（1. *College of Information and Electrical Engineering*, *China Agricultural University*, *Beijing* 100083, *China*; 2. *National Innovation Center for Digital Fishery*, *China Agricultural University*, *Beijing* 100083, *China*; 3. *Beijing Engineering and Technology Research Centre for the Internet of Things in Agriculture*, *Beijing* 100083, *China*; 4. *School of Information Science and Technology*, *Beijing Forestry University*, *Beijing* 100083, *China*）

Abstract: Convolutional neural network models have different advantages and disadvantages, it is becoming more and more difficult to select an appropriate convolutional neural network model in an actual fish identification project. The identification of underwater fish is a challenge task due to varies in illumination, low contrast, high noise, low resolution and sample imbalance between each type of image from the real underwater environment. In addition, deploying models to mobile devices directly will reduce the accuracy of the model sharply. In order to solve the above problems, Fish Recognition Ground-Truth dataset was used to training model in this study, which is provided by Fish4Knowledge project from University of Edinburgh. It contains 27, 370 images with 23 fish species, and has been labeled manually by marine biologists. AlexNet, GoogLeNet, ResNet and DenseNet models were selected initially according to the characteristics of real-time underwater fish identification task, then a comparative experiment was designed to explore the best network model. Random image flipping, rotation and color dithering were used to enhance data based on ground-truth fish dataset in response to the limited number of underwater fish images. Considering that there was a serious imbalance in the number of samples in each category, the label smoothing technology was used to alleviate model overfitting. The Ranger optimizer and Cosine learning rate attenuation strategy were used to further improve the training effect of the models. The accuracy and recall rate information of each model were recorded and counted. The results showed that, the accuracy and recall rate of the fish recognition model based on DenseNet reached 99.21% and 96.77% in train set and validation set respectively, its F_1 value reached 0.9742, which was the best model obtained in the experiment. Finally, a remote fish identification system was designed based on Python language, in this system the model was deployed to linux server and the Android APP was responsible for uploading fish images via http to request server to identify the fishes and displaying the identification information returned by server, such as fish species, profiles, habits, distribution, etc. A set of recognition tests were performed on real Android phone and the results showed that in the same local area net the APP could show fish information rapidly and exactly within 1 s.

Key words: fish identification model; CNN; model evaluation; Android; Ground-Truth; real-time identification system